FINDING NON-OBVIOUS PROFILES BY USING ANT-ALGORITHMS

Thomas Ambrosi
Information Technology, Nortel GmbH, D-60549 Frankfurt/Main, Germany

Sascha Kaufmann
FSTC/ILIAS/MINE; Université du Luxembourg, 6, rue R. Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg

Keywords: User interest profiles, Recommendations, Ant-algorithms.

Abstract: Visitors on a website are usually on their own when they are moving around. First time visitors are especially guessing where to find the information they are looking for. In this paper we will show a way to combine the concepts of non-obvious profiles and ant-algorithms to come up with a set of paths, which tries to cover the user's interests in a proper way. These paths can be used to give recommendations to visitors. While the profiles help to get a better understanding of the users' interests, the concept of ant-algorithms is employed to determine recently and frequently used paths to lead the user to the desired information. We explain the basic idea of our approach, the current state of the prototype realized and some first results.

1 INTRODUCTION

There are many factors that influence a user's experience when visiting a website (e.g. GUI, content, structure). In this work we concentrate on a web-site's structure. We focus on content-directed users, which are looking for interesting information. For this reason, we are assuming, that visitors want to minimise the number of pages to be visited during their attendance which revise stress and gives a good surfing experience. We want to achieve this by performing a personalised optimisation.

The work is partitioned into the following sections: In section 2 we describe the related work. In section 3 we present our new approach. Section 4 contains first results, while section 5 gives a summary and an outlook to further work.

2 RELATED WORK

Our approach combines two algorithms: the NOP-Algorithm and the concept of ant-algorithms.

The concept of "Non-Obvious Profiles (NOP)" was initially designed to measure the level of interests in certain topics of web portal users over a period of time (Mushtaq et al., 2004). In an initial process, every page of a website is assigned one or more combinations of weighted topics, that indicate how well a topic represents the page's content. This is used later to calculate an interest profile each time a person visits the website. The algorithm is doing this by only taking into account how long the user stays at each page during the session. As an adjusting mechanism, the user is asked from time to time about his current interests. The results are used to calibrate the formerly calculated profiles. The algorithm was later extended by adding the concepts of areas and actions to improve the profile's quality (Hoebel et al., 2006).

The ant-algorithm is based on experiments done on ants (e.g. (Goss et al., 1989), (Jeanson et al., 2003)), which observed them in their natural environment and how they communicated via pheromones. Pheromones are scents that are secreted by ants. They create a kind of path of scent that other ants can follow. Over time, depending on the kind of surface, the scent vanishes if it is not renewed. Pheromones are employed for navigation during different actions (e.g. searching for food). In computer-science artificial ant-algorithms are mainly used for optimisation problems, as the "Travelling Salesman Problem (TSP)" (Bonabeau et al., 1999).

Ambrosi T. and Kaufmann S.
FINDING NON-OBVIOUS PROFILES BY USING ANT-ALGORITHMS.
DOI: 10.5220/0001842806740679
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

3 OUR NEW APPROACH

As presented, our approach combines the NOP's characteristic of interesting profiles with the ant approach of handling paths. We use the characteristic of a positive reinforcement of short ways, which is presented by the natural trace of pheromones.

Like in the TSP, we model the covered distance as a directed graph. Adding the according pheromone assignment to the appropriate accumulations will then become visible.

In our work, we refer to the "Adaptive Colony System (ACS)-TSP algorithm" (Bonabeau et al., 1999). The ant covers a distance during its search for food and its return journey. It marks its path with a trace of one or more different pheromones. Every ant is able to orient itself using these traces. The trace's strength will be important when the ant's path splits. In most cases, the ant will follow the stronger trace, but its decision can be influenced by other parameters, too.

In our model a visitor adopts the ants role and has the ability to dispense and recognise a special pheromone for each topic (i.e. interests of the visitor). The released amount of a pheromone is related to the time a visitor is on a page and the topics (along with the weights), which are related to this page. It is also decreasing with any additional visited page. The ant is able to recognise existing traces of pheromones and store them as a non-obvious profile. These traces can be given by the website administrator or by other visitors.

Applications of our combined approach could be to indicate visitors' interests in a faster way and suggest web pages that might cover their interest profiles with a high probability. We use the pheromone trace to analyse the website for typical patterns of paths taken by visitors, based on their interests.

3.1 Distances

As shown in (Traniello, 1989), ants do not use arbitrary ways to get their food. Usually they try to minimise the distance. We consider this by introducing a factor *PherDist*, which is steadily decreasing with each additional page. As already mentioned we assume, that after a large number of requests, the user is not interested anymore in the content (content-directed visitor). We describe this scenario as follows:

- The first pages are very important
- The importance is constantly decreasing

We have experimented with two different distance functions to influence a user profile. We are using equation (1) to express an exponential decreasing, where l is the length of the path, i.e. number of pages, and f is a factor controlling the aging rate.

$$PherDist(l) = e^{f*l^2} \text{where } f \leq 0 \qquad (1)$$

Alternatively, we are suggesting a function with a linear aging rate that prunes the influence of later pages at one point:

$$PherDist(l) = \begin{cases} -\frac{l}{f}+1 & \text{where } 0 < l \leq f \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

In practice, these functions depend on the purpose of the website. Other functions can be considered as long as they satisfy the condition (3):

$$0 \leq PherDist(l) \leq 1, \quad l = 1, \ldots, l_{max} \qquad (3)$$

3.2 Aging of Pheromones

While the information of knowing where to find the food is very important, it is as important to know when a path was last used. The assumption here is to recommend a path that was taken by a visitor with the same interests in the recent history. Here we use the fact that the strength of the pheromones are steadily decreasing over time (Jeanson et al., 2003). After each iteration the pheromones' strengths are decreasing by a factor *PherAge*. Similar to (Bonabeau et al., 1999) we use an exponential decreasing function (4) for our experiments, where w_k is the visitor's current value of interest for topic k, f is a factor and t is the number of iteration between the last computation of the visitor's profile and the current one.

$$PherAge(t) = w_k * e^{f*t^2} \text{ where } f \leq 0 \qquad (4)$$

As an alternative we used function (5) in our experiments which has a different behaviour.

$$PherAge(t) = f * w_k^t \text{ where } f \leq 0 \qquad (5)$$

The definition of an iteration is very important and therefore the understanding of 'recent past'. In our work we assume that all actions that occur in an iteration take place simultaneously. In practice it highly depends on the amount of incoming users in a time span. An appropriate value for $PherAge(t)$ should satisfy condition (6):

$$0 \leq PherAge(t) \leq 1, \quad t = 1, \ldots, t_{max} \qquad (6)$$

3.3 Calculating the Extended NOP

We consider a website as a directed graph $G = (N, E)$, where N is the set of all pages. The directed edges E are all v_{ij} where j is directly reachable from i. We assign all topics and their weights of page P_i to the edges v_{ij} and the pheromones are represented by the topics.

A user's visit is represented as a sequence of edges $v_1, ..., v_m$. We are then able to calculate the k-pheromone's contribution τ^k, to the non-obvious profile's value w_k for topic k by using (7), where $duration(v_i)$ is the time the user stayed on Page P_i. Note that using the duration of the whole session normalises the pheromones' sums.

$$\Delta\tau^k = \frac{\sum_{i=1}^{m} duration(V_i) \cdot PherDist(i) \cdot v_i(\tau^k)}{\sum_{j=1}^{m} duration(V_j)} \quad (7)$$

In the next step the pheromones sum is added to the existing non-obvious user profile. As the pheromones' values decrease over time, we calculate the non-obvious profile as follows (8):

$$w_k = w'_k \cdot PherAge(Date(w_k)) + \sum_{i=1}^{SessionCount} PherAge(SessionDate(i)) \cdot \Delta\tau^k(Session_i) \quad (8)$$

Here, w_k is the value that represents the users interest in topic k. $Date$ and $SessionDate$ represent the time and the number of iterations respecting since the last computation of the non-obvious profile.

4 RESULTS

We experimented with data from three existing websites, however, in this paper we present only the website with the best results. The reason for this is that we had to assign topics manually by reading the content. This website was relatively 'small' (approx. 400 pages) and static. It allowed us to do the topic-weight assignment in an accurate way, while the other two websites were either very large and/or already changed by the time we did the assignment.

4.1 Testing Environment

The websites were not designed to support the concept of non obvious profiles. A feedback mechanism, to calibrate the visitor's interests, was also not given. We used the anonymized log file of a commercial website with removed known robots' entries (Cooley et al., 1999). We then defined ten topics, which described in our understanding the website in a good way and manually assigned the relevant topics to every page. Note that this process is very subjective and is highly influenced by the person who is doing the assignments. Table (1) tabulates our test data parameters, whilst Figure (1) gives an overview of the distribution of the number of pages per session.

Table 1: Testing environment parameters.

Parameter	*Value*
Observed Period	09/27/2005 - 12/27/2005
# Entries	57 785
# Sessions	6 227
# Sessions ≥ 3 requests	3 887
∅ Session Length (≥ 3)	14.115
Filters	Elimination of known robots
# Topics	10

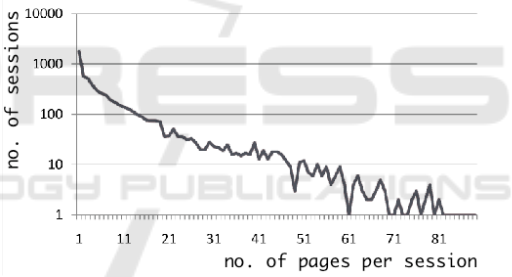

Figure 1: Distribution no. of pages vs. no. of session.

We determined the exposure time Δt for each page by comparing the times when the web pages were requested (log file). We calculated Δt using (9), where t_1 is the point in time when page 1 was sent, and t_0 is the point in time when page 0 was sent, with $t_1 \geq t_0$.

$$\Delta t = t_1 - t_0 \quad (9)$$

To identify a user during a session we used the ip-address or, in case of existence, a session id. Furthermore we used a session time-out of 24 minutes for all pages, because it was already defined on one website.

Given this data, we came up with a set of click streams. A click stream represents a directed graph $G_v = (N, E)$, where G_v is a set of visited pages. The directed edges are all v_{mn} with n being the page following m. Finally, the exposure time is added as an attribute to the outgoing edges.

4.2 Test Results

In the following we show the results for the obtained profiles by applying the algorithm. To compare the different results in a better way, we computed the maximum and the average values for each pheromone/topic based on the used iteration time span. The average pheromone values were computed by all profiles with more than one session. The reason for this was not to influence these values by too many 'trivial' NOPs and to get a better understanding of multiple visits.

For getting a feeling of the level of influence of the *PherAge*-function we experimented with two different functions. We used the values $f = 0.8, f = 0.9, f = 0.99$ in conjunction with equation (5) while the values $f = -0.002, f = -0.00025$ indicates that we used equation (4).

4.2.1 Results Iteration Size 24 Hours

Table 2 shows the average values in the results for an iteration time span of 24 hours. This values seem good, but a closer look at the maximum values for the topics in Table 3 shows that the values can rise very high in comparison with the original NOP-approach. This happens if a visitor comes back within 24 hours (multiple sessions) there will be a direct addition of the iteration's profile values.

There are two disadvantages with this solution: it makes it difficult to compare the values and we can easily become dependent on a few users. Promising modification of the algorithm can be to decrease the interval's time span or to satiate the pheromones' values if there are multiple sessions of the same visitor during an iteration.

Table 2: NOPs average values (24 h).

Avg.	f=0.8	f=0.9	f=0.99	f=-0.002	f=-2.5*10^{-4}
P 01	0.25	0.26	0.30	0.28	0.30
P 02	0.07	0.08	0.09	0.08	0.09
P 03	0.11	0.11	0.13	0.12	0.13
P 04	0.09	0.10	0.12	0.11	0.12
P 05	0.07	0.07	0.09	0.08	0.09
P 06	0.17	0.17	0.20	0.18	0.20
P 07	0.17	0.18	0.20	0.18	0.20
P 08	0.56	0.58	0.64	0.60	0.64
P 09	0.69	0.71	0.79	0.74	0.80
P 10	0.52	0.54	0.62	0.58	0.63

4.2.2 Results Iteration Size 6 Hours

In the next step we reduced the iterations' time span to 6 hours. There were no real improvement, the values were nearly the same as in the 24 hours iteration interval (Tables 4 and 5).

Table 3: NOPs maximum values (24 h).

Max.	f=0.8	f=0.9	f=0.99	f=-0.002	f=-2.5*10^{-4}
P 01	1.57	1.57	1.77	1.71	1.81
P 02	0.59	0.59	0.59	0.59	0.59
P 03	1.12	1.12	1.12	1.12	1.12
P 04	0.88	0.88	0.88	0.88	0.88
P 05	0.98	0.98	0.98	0.98	0.98
P 06	1.35	1.35	1.35	1.35	1.35
P 07	1.32	1.32	1.32	1.32	1.32
P 08	3.19	3.34	3.81	3.48	3.90
P 09	2.65	2.90	4.77	3.28	4.85
P 10	2.72	2.72	2.72	2.72	2.72

This indicates a constant interest in a topic and that visitors with multiple sessions within 6 hours, being constantly interested in the same kind of topics.

Table 4: NOPs average values (6 h).

Avg.	f=0.8	f=0.9	f=0.99	f=-0.002	f=-2.5*10^{-4}
P 01	0.24	0.25	0.28	0.26	0.27
P 02	0.07	0.07	0.08	0.07	0.08
P 03	0.10	0.11	0.12	0.11	0.12
P 04	0.08	0.09	0.11	0.09	0.10
P 05	0.06	0.07	0.08	0.07	0.08
P 06	0.16	0.16	0.18	0.17	0.18
P 07	0.16	0.17	0.18	0.17	0.18
P 08	0.54	0.56	0.60	0.57	0.59
P 09	0.65	0.67	0.74	0.70	0.72
P 10	0.49	0.51	0.58	0.54	0.57

Table 5: NOPs maximum values (6 h).

Max.	f=0.8	f=0.9	f=0.99	f=-0.002	f=-2.5*10^{-4}
P 01	1.57	1.57	1.64	1.57	1.69
P 02	0.59	0.59	0.59	0.59	0.59
P 03	1.12	1.12	1.12	1.12	1.12
P 04	0.88	0.88	0.88	0.88	0.88
P 05	0.98	0.98	0.98	0.98	0.98
P 06	1.35	1.35	1.35	1.35	1.35
P 07	1.24	1.25	1.31	1.32	1.32
P 08	3.19	3.34	3.47	3.48	3.48
P 09	2.44	2.70	3.62	2.97	3.03
P 10	2.62	2.45	2.69	2.71	2.72

4.2.3 Results Iteration Size 24 Hours with Pheromones' Limitations

To avoid the effect described above, we limited the amount of pheromones that a user is allowed to spend within an iteration. We obtained this by adding the arithmetic mean of the pheromone's values instead of adding the values from each session (limited adjustment). With this method we got stable NOPs (Tables 6 and 7).

However, there remains the challenge to interpret the pheromone's strength in the profiles, because the values are still not normalised.

Table 6: NOPs average values / limited (24 h).

avg	f=0.8	f=0.9	f=0.99	f=-0.002	f=-2.5*10^{-4}
*P*01	0.14	0.15	0.18	0.16	0.18
*P*02	0.05	0.05	0.06	0.05	0.06
*P*03	0.07	0.07	0.09	0.08	0.09
*P*04	0.06	0.06	0.08	0.07	0.09
*P*05	0.04	0.05	0.06	0.05	0.06
*P*06	0.09	0.10	0.12	0.11	0.12
*P*07	0.10	0.10	0.12	0.11	0.12
*P*08	0.29	0.30	0.34	0.32	0.35
*P*09	0.36	0.38	0.44	0.41	0.45
*P*10	0.29	0.31	0.37	0.34	0.38

Table 7: NOPs max. values / limited (24 h).

max	f=0.8	f=0.9	f=0.99	f=-0.002	f=-2.5*10^{-4}
*P*01	1.34	1.50	1.72	1.71	1.74
*P*02	0.49	0.49	0.49	0.49	0.49
*P*03	0.92	0.92	0.92	0.92	0.92
*P*04	0.88	0.88	0.88	0.88	0.88
*P*05	0.95	0.95	0.95	0.95	0.95
*P*06	0.95	0.95	0.95	0.95	0.95
*P*07	0.95	0.95	0.95	0.95	0.98
*P*08	1.35	1.41	1.56	1.47	1.64
*P*09	1.23	1.34	1.90	1.53	1.90
*P*10	1.43	1.80	2.57	2.30	2.64

4.2.4 Environment for Predicting the Next Page

The primary goal of the algorithm is to lead visitors to the content they are interested in. Because only log-files were available, we did a first test by measuring how accurate we can predict visitors' behaviour based on the interest values we calculated.

We performed two series of experiments. In the first serie we split the original set of 6000 sessions into the training and test sets which contained 4,000 and 2,000 sessions respectively. In the second serie we split it into 5000 and 1000 sessions. To avoid a high influence by short sessions, we only took session with $3 \leq session\_length \leq 50$. This reduced the number to 2219 and 3013 sessions in the training sets.

We used these sessions to simulate visits on the web-site. During the runs we calculated the visitors' current profile of interest at every 5th, 7th, and 9th step (page). We ranked the outgoing links of the page in order to compare these profiles with the links the visitors follow during the session. This approach was chosen to proof the assumption that the knowledge of the visitor's interests will help to get good results To get a point of reference (benchmark), we compared these results to a basic counting algorithm. We initialised the graph like in the previous approach. But instead of calculating pheromone values, we counted for each link how often it was taken by the visitors. The obtained numbers were used during the simulation to rank the outgoing links and predict the next page. Table 8 gives an overview of the parameters for this test.

Table 8: Testing environment parameters.

Parameter	*Value*
# Sessions	6 000
# Topics	10
Iteration Size	24 h
# Training Sessions (4000)	2 419
# Training Sessions (5000)	3 013

4.2.5 Test Results for Predicting the Next Page

In the following, we present the results for the different runs. Tables 9 and 10 show the results for the smaller training set while tables 11 and 12 presents the results for the larger one. The rank values indicate the rate of success based on the number of predictions for the next page. For example: "Rank 2, 5th Page" in table 9 indicates that we were successful in nearly 50% of all cases to predict the 6th page in a session, if we are allowed to give two suggestions.

Table 9: Results (ANOP / Training Session (4000)).

ANOP.	5th Page	7th Page	9th Page
*Rank*1	336 (40.2%)	279 (41.3%)	248 (43.6%)
*Rank*2	419 (50.1%)	345 (51.0%)	302 (53.1%)
*Rank*3	484 (57.9%)	398 (58.9%)	340 (59.8%)
*Rank*4	521 (62.2%)	433 (64.0%)	375 (65.9%)
*Rank*5	556 (66.4%)	460 (68.0%)	402 (70.7%)
...	...	...	...
ALL	836 (100%)	676 (100%)	569 (100%)

Both algorithms provide relatively good results. A larger set of training data leads, as expected, to a better result in the verification process. But it seems that we can better predict the next pages by following the majority of the previous visitors.

What are the conclusions from these results? On the one hand it seems that the algorithm is not suitable to predict a visitor's next. On the other hand, in a productive system the algorithm would be used to suggest a number of pages that will fit in the best way to a profile. This can be every page on a web-site and is not limited to the next reachable pages. It could also be that the property of the algorithm to 'forget' information after some time is turning in this case into a disadvantage. As mentioned above, the results depend on many parameters. To examine this behaviour in a deeper way will be part of our future work.

Table 10: Results (Count / Training Session (4000)).

Count	5th Page	7th Page	9th Page
*Rank*1	368 (44.0%)	308 (45.6%)	265 (46.6%)
*Rank*2	457 (54.7%)	377 (55.8%)	329 (57.8%)
*Rank*3	513 (61.4%)	424 (62.7%)	371 (65.2%)
*Rank*4	562 (67.2%)	457 (67.6%)	401 (70.8%)
*Rank*5	601 (71.9%)	497 (73.5%)	420 (73.8%)
...	...	...	...
ALL	836 (100%)	676 (100%)	569 (100%)

Table 11: Results (ANOP / Training Session (5000)).

ANOP.	5th Page	7th Page	9th Page
*Rank*1	115 (41.3%)	139 (45.9%)	121 (47.1%)
*Rank*2	194 (51.7%)	168 (55.5%)	147 (57.2%)
*Rank*3	225 (60.0%)	190 (62.7%)	163 (63.4%)
*Rank*4	236 (62.9%)	201 (66.3%)	173 (67.3%)
*Rank*5	251 (66.9%)	213 (70.3%)	180 (70.4%)
...	...	...	...
ALL	375 (100%)	303 (100%)	257 (100%)

Table 12: Results (Count / Training Session (5000)).

Count	5th Page	7th Page	9th Page
*Rank*1	183 (48.8%)	156 (51.5%)	130 (50.6%)
*Rank*2	242 (64.5%)	196 (64.7%)	162 (63.0%)
*Rank*3	272 (72.5%)	223 (73.6%)	180 (70.0%)
*Rank*4	298 (79.5%)	236 (77.8%)	194 (74.5%)
*Rank*5	313 (83.5%)	253 (83.5%)	206 (80.2%)
...	...	...	...
ALL	375 (100%)	303 (100%)	257 (100%)

Both algorithms provide relatively good results. A larger set of training data leads, as expected, to a better result in the verification process. But it seems that we can better predict the next pages by following the majority of the previous visitors.

What are the conclusions from these results? On the one hand it seems that the algorithm is not suitable to predict a visitor's next. On the other hand, in a productive system the algorithm would be used to suggest a number of pages that will fit in the best way to a profile. This can be every page on a web-site and is not limited to the next reachable pages. It could also be that the property of the algorithm to 'forget' information after some time is turning in this case into a disadvantage. As mentioned above, the results depend on many parameters. To examine this behaviour in a deeper way will be part of our future work.

5 CONCLUSIONS

We have shown how the combination of the NOP and ant-algorithm approach can help to derive individual profiles once the topics-weights assignment and the possibility to recognise users are taken into account. Therefore it can be used to support the individual user. Our current experiments are based on log files and our subjective assignment of topics and their weights to every page.

The most important future work will be to improve the algorithm to produce normalised values for a better comparison of the different models. It is also important to implement feedback functionality and a recommendation system that supports the visitor in an active way. We would also like to automate the process of topic assignment to make it less subjective.

ACKNOWLEDGEMENTS

This work was funded by the Fonds National de la Recherche Luxembourg (FNR). We thank Prof. C. Schommer (Université du Luxembourg), Prof. R. Zicari (Goethe-University, Frankfurt/Main, Germany) and Nortel GmbH.

REFERENCES

Ambrosi, T. (2007). Erstellung und Analyse von nicht offensichtlichen Profilen mit Hilfe von Ameisenalgorithmen.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence – From Natural to Artifical Systems.* Oxford.

Cooley, R., Mobasher, B., and Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. In *Knowledge and Information Systems*. Springer-Verlag.

Goss, S., Aron, J., Deneubourg, J., and Pasteelsi, J. (1989). Self-organized shortcuts in the argentine ant. In *Naturwissenschaften 76*, pages 579–581. Springer-Verlag.

Hoebel, N., Kaufmann, S., Tolle, K., and Zicari, R. (2006). The design of gugubarra 2.0: A tool for building and managing profiles of web users. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*.

Jeanson, R., Ratnieks, F. L. W., and Deneubourg, J.-L. (2003). Pheromone trail decay rates on different substrates in the pharaohs ant, *Monomorum pharaonis*. In *Physiological Entomology*, volume 28, pages 192–198.

Mushtaq, N., Tolle, K., Werner, P., and Zicari, R. (2004). Building and evaluating nonobvious user profiles for visitors of web sites. In *e-Commerce Technology, 2004. CEC 2004. Proceedings. IEEE International Conference on*.

Traniello, J. F. A. (1989). Foraging strategies of ants. In *Annual Reviews Entomology*, volume 34, pages 191–210.